We Claim:

1. A method of representing type information via objects of classes in a class hierarchy, wherein the class hierarchy comprises at least one class and a plurality of sub-classes for representing different type classifications, the method comprising:

instantiating one or more objects of one or more of the sub-classes of the hierarchy, wherein the one or more sub-classes represent classifications of types; and

storing information in the one or more objects.

2. The method of claim 1 wherein the one or more objects represent type information of a variable in software during compilation of the software.

3. The method of claim 1 wherein at least one of the objects comprises information for a size of a type represented by the object.

4. The method of claim 1 wherein at least one of the one or more sub-classes inherits from an abstract type that wraps an externally defined type, the abstract type providing a mapping from a typed intermediate language to original source code.

5. The method of claim 1 wherein at least one of the one or more sub-classes represents container types.

6. The method of claim 1 wherein at least one of the one or more sub-classes represents pointer types.

7. The method of claim 1 wherein at least one of the one or more sub-classes represents function types.

8. The method of claim 1 wherein at least one of the one or more sub-classes represents unmanaged array types.

9. The method of claim 1 wherein at least one of the one or more sub-classes represents class types.

10. The method of claim 1 wherein at least one of the one or more sub-classes represents managed array types.

11. The method of claim 1 wherein at least one of the one or more sub-classes represents struct types.

12. The method of claim 1 wherein at least one of the one or more sub-classes represents interface types.

13. The method of claim 1 wherein at least one of the one or more sub-classes represents enumerated types.

14. The method of claim 1 wherein at least one of the one or more sub-classes represents primitive types.

15. The method of claim 14 wherein at least one of the sub-classes representing primitive types represents the following types: int, float, and void.

16. The method of claim 14 wherein at least one of the sub-classes representing primitive types can represent an unknown type.

17. The method of claim 14 wherein at least one of the sub-classes representing primitive types is extensible to represent one or more additional primitive types.

18. The method of claim 1 wherein at least one of the one or more sub-classes is defined from the group consisting of: 'ContainerType', 'PtrType', 'FuncType', 'ClassType', 'StructType', 'InterfaceType', and 'EnumType'.

19. The method of claim 1 wherein at least one of the one or more sub-classes is defined as 'PrimType'.

20. A computer-readable medium having a software program thereon, the program comprising:

program code for defining a programming class as 'PrimType';

program code for associating a size with instances of the 'PrimType' class; and

program code for associating a kind of type with instances of the 'PrimType' class.

21. The computer-readable medium of claim 20 wherein the size represents a size of a machine representation of a value.

22. The computer-readable medium of claim 20 wherein the program code for associating a size with instances of the 'PrimType' class defines the size as 'BitSize'.

23. The computer-readable medium of claim 20 wherein the kind of type represents a type classification.

24. The computer-readable medium of claim 20 wherein the program code for associating a kind of primitive type with instances of the 'PrimType' class defines the kind of type as 'PrimTypekind'.

25. The computer-readable medium of claim 20 further comprising program code for associating a type of size with instances of the 'PrimType' class.

26. The computer-readable medium of claim 25 wherein the program code for associating a type of size with instances of the 'PrimType' class defines the type of size as 'SizeKind'.

27. The computer-readable medium of claim 25 wherein the type of size can comprise actual, symbolic, or unknown types of size.

28. The computer-readable medium of claim 20 wherein the class 'PrimType' represents a plurality of types, the plurality of types comprising int, float, unknown, void, condition code, and unsigned int types.

29. A computer-readable medium having computer-executable instructions for implementing a method for representing type classifications, the method comprising:

describing a type classification for representing container types of a plurality of programming languages, wherein the type classification is described as 'ContainerType'.

30. The computer-readable medium of claim 29, wherein the method further comprises:

describing a type classification for representing pointer types of a plurality of programming languages, wherein the type classification is described as 'PtrType'.

31. A method of programmatically defining a type representation, the method comprising:

defining a base class;

defining a plurality of classes hierarchically below the base class, wherein the classes represent at least pointer types, container types and function types of a plurality of programming languages.

32. The method of claim 31 further comprising defining a plurality of classes hierarchically below the class representing container types, wherein the plurality of classes represent at least class types, struct types, interface types, and enumerated types of a plurality of programming languages.

33. The method of claim 32 further comprising defining a class hierarchically below the class representing class types, wherein the class represents unmanaged array types of a plurality of programming languages.

34. The method of claim 31 further comprising defining a class hierarchically one of the plurality of classes, wherein the class represents primitive types of a plurality of programming languages.

35. The method of claim 34 wherein the class further represents an unknown type.

36. A computer-readable medium having a software program thereon, the program comprising:

program code for defining a programming class as 'ContainerType', wherein an object of class 'ContainerType' is a type representation in an intermediate language for container types in a section of code written in one of a plurality of programming languages;

program code for defining a programming class as 'PtrType', wherein an object of class 'PtrType' is a type representation in an intermediate language for pointer types in a section of code written in one of a plurality of programming languages;

program code for defining a programming class as 'FuncType', wherein an object of class 'FuncType' is a type representation in an intermediate language for function types in a section of code written in one of a plurality of programming languages;

program code for defining a programming class as 'ClassType', wherein an object of class 'ClassType' is a type representation in an intermediate language for class types in a section of code written in one of a plurality of programming languages;

program code for defining a programming class as 'StructType', wherein an object of class 'StructType' is a type representation in an intermediate language for struct types in a section of code written in one of a plurality of programming languages;

program code for defining a programming class as 'InterfaceType', wherein an object of class 'InterfaceType' is a type representation in an intermediate language for interface types in a section of code written in one of a plurality of programming languages; and

program code for defining a programming class as 'EnumType', wherein an object of class 'EnumType' is a type representation in an intermediate language for enumerated types in a section of code written in one of a plurality of programming languages.

37. The computer-readable medium of claim 36 further comprising program code for defining a programming class as 'PrimType', wherein an object of class 'PrimType' is a type representation in an intermediate language for primitive types in a section of code written in one of a plurality of programming languages.

38. The computer-readable medium of claim 36 further comprising program code for associating a size with an object of any class.

39. The computer-readable medium of claim 36 further comprising program code for associating a kind of type with an object of any class.